University of Greifswald

Faculty of Mathematics and Natural Sciences

Institute for Geographie and Geology

Code Documentation

Full Documentation of AlphaBetaSquared Class for Python

Greifswald April 8, 2025

Contents

1 Overview

The `AlphaBetaSquared` class is a Python utility designed for statistical analysis and visualization of particle size distributions from CSV data files. It computes key statistical parameters (α and β^2), fits log-normal distributions, and generates plots to visualize both size distributions and kinetic analysis after Eberl et al., 1998 (alpha-beta squared plots).

The class supports flexible data loading, customizable plotting, and data export functionalities, making it suitable for both interactive use (e.g., Jupyter notebooks) and batch processing.

2 Class Description

Class Definition

```
class AlphaBetaSquared():
```

Purpose

The `AlphaBetaSquared` class provides methods to:

- Load data from one or more CSV files.
- Calculate statistical metrics (α, β, maximum values, and log-normal fits).
- Visualize particle size distributions and α vs. β^2 plots.
- Export computed data to CSV files.

Attributes

- `self.data`: Dictionary storing loaded CSV data.
- `self._alpha`: Dictionary of computed α values.
- `self._beta`: Dictionary of computed β values.
- `self._table_max`: Dictionary of maximum values per column.
- `self._lognorm_fits`: Dictionary of log-normal fit parameters.
- `self.auto_display`: Boolean to control automatic plot display (default: `False`).
- `self.alphabeta_scale`: Boolean to scale α-β^2 plot limits (default: `False`).
- `self.save_plot`: Boolean to save plots to files (default: `False`).
- `self.plot_config`: Dictionary for plot formatting (partially implemented).
- `self.__exportable_attributes`: Tuple of attributes available for export.

Dependencies

Package	Version	Installation Command
Python	≥3.8	`conda install python=3.9.2`
NumPy	1.23.x	`pip install numpy==1.23.4`
SciPy	1.9.x	`pip install scipy==1.9.1`
Matplotlib	3.6.x	`pip install matplotlib==3.6.2`
Pandas	2.1.x	`pip install pandas==2.1.4`

3 Methods

3.1 `__init__`

```
def __init__(self, *args, plot_config=None)
```

Description: Initializes the class by loading data and computing statistical metrics.
Parameters:

- `*args`: Variable-length argument list of CSV file paths or a single list of paths.
- `plot_config`: Optional dictionary to customize plot appearance (default: `None`).

Behavior: Loads data, calculates α, β, maximum values, and log-normal fits, and sets up plotting attributes.

3.2 `_load_data`

```
def _load_data(self, args)
```

Description: Loads data from CSV files into a dictionary.
Parameters:

- `args`: Tuple of file paths or a single list of paths.

Returns: Dictionary with keys as table names (derived from filenames) and values as `pandas.DataFrame` objects.

3.3 `set_auto_display`

```
def set_auto_display(self, x)
```

Description: Sets the `auto_display` attribute.
Parameters:

- `x`: Boolean value.

3.4 `set_save_plot`

```
def set_save_plot(self, x)
```

Description: Sets the `save_plot` attribute.
Parameters:

- `x`: Boolean value.

3.5 `set_alphabeta_scale`

```
def set_alphabeta_scale(self, x)
```

Description: Sets the `alphabeta_scale` attribute.
Parameters:

- `x`: Boolean value.

3.6 `calc_alpha`

```
def calc_alpha(self, x)
```

Description: Calculates the mean of the natural logarithm of the input data.
Parameters:

- `x`: Numeric array-like object.

Returns: Float representing α.

3.7 `calc_beta`

```
def calc_beta(self, x)
```

Description: Calculates the variance of the natural logarithm of the input data.
Parameters:

- `x`: Numeric array-like object.

Returns: Float representing β.

3.8 `show`

```
def show(self)
```

Description: Displays all active Matplotlib plots.

3.9 `get_data`

```
def get_data(self)
```

Returns: The `self.data` dictionary.

3.10 `get_alpha`

```
def get_alpha(self)
```

Returns: The `self._alpha` dictionary.

3.11 `get_beta`

```
def get_beta(self)
```

Returns: The `self._beta` dictionary.

3.12 operate_on

```
def operate_on(self, data_dict, stat_func, result_attr, *args, **kwargs)
```

Description: Applies a statistical function to each column of each table in a data dictionary and stores results.
Parameters:

- `data_dict`: Dictionary of data tables.
- `stat_func`: Function to apply to each column.
- `result_attr`: Attribute name to store results.
- `*args`, `**kwargs`: Additional arguments for `stat_func`.

3.13 calc_distribution

```
def calc_distribution(self, bins=50, density=True, *args, **kwargs)
```

Description: Calculates histograms for particle size distributions.
Parameters:

- `bins`: Integer or array specifying bin edges (default: 50).
- `density`: Boolean to normalize histogram (default: `True`).
- `*args`, `**kwargs`: Additional arguments passed to histogram calculation.

3.14 plot_distribution

```
def plot_distribution(self, table_column_map=None)
```

Description: Plots particle size distributions with log-normal fits.
Parameters:

- `table_column_map`: Optional dictionary mapping table names to column lists (default: `None`).

Returns: List of `matplotlib.figure.Figure` objects.
Notes: Automatically calculates distributions if not already computed. Supports saving plots if `self.save_plot` is `True`.

3.15 plot_alphabeta

```
def plot_alphabeta(self, *args, **kwargs)
```

Description: Generates an α vs. β^2 scatter plot with regression lines and kinetic regime annotations.
Parameters:

- `*args`: Optional table names or dictionary specifying tables to plot.
- `**kwargs`: Reserved for future extensions.

Returns: Tuple of `matplotlib.figure.Figure` and `matplotlib.axes.Axes` objects.

3.16 save_data

```
def save_data(self, attr, *args, **kwargs)
```

Description: Saves a specified attribute's data to CSV files.
Parameters:

- `attr`: String name of the attribute to save.
- `*args`, `**kwargs`: Reserved for future extensions.

3.17 export_data

```
def export_data(self, *args, **kwargs)
```

Description: Exports specified or all exportable attributes to CSV files.
Parameters:

- `*args`: Optional attribute names to export.
- `**kwargs`: Reserved for future extensions.

4 Usage Examples

Basic Usage

```
obj = AlphaBetaSquared('data1.csv', 'data2.csv')
obj.plot_distribution()
obj.plot_alphabeta()
obj.export_data()
```

Custom Plotting

```
obj = AlphaBetaSquared('data.csv')
obj.set_auto_display(False)
figs = obj.plot_distribution({'table1': ['col1', 'col2']})
figs[0].get_axes()[0].set_title('Custom Title')
obj.show()
```

5 Notes

- The `plot_config` attribute is partially implemented and marked as a TODO.
- Invalid tables or columns in plotting methods are skipped with warnings.
- Set `auto_display` to `False` in Jupyter notebooks when manipulating returned figures.

References

Eberl, D., Drits, V., & Srodon, J. (1998). Deducing growth mechanisms for minerals from the shapes of crystal size distributions. *American journal of Science*, *298*(6), 499–533.

List of Figures

List of Tables